Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems ••• (••••) •••–•••
FUZZY sets and systems
www.elsevier.com/locate/fss

T-generable indistinguishability operators and their use for feature selection and classification

Eva Armengol [a,*], Dionís Boixader [b], Àngel García-Cerdaña [a,c], Jordi Recasens [b]

[a] *Artificial Intelligence Research Institute (IIIA–CSIC), Campus de Bellaterra, E-08193 Bellaterra, Catalonia, Spain*
[b] *Architecture Technology Department, ETSAV-UPC, Sant Cugat del Vallès, Catalonia, Spain*
[c] *Information and Communication Technologies Department, University Pompeu Fabra, Tànger, 122-140, E-08018 Barcelona, Catalonia, Spain*

Received 4 December 2017; received in revised form 1 March 2018; accepted 2 March 2018

Abstract

T-generable indistinguishability operators are operators E that can be expressed in the form $E = T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$, where T is a t-norm and E_μ is the fuzzy relation generated by the fuzzy subset μ. In this paper we analyse their relation with powers with respect to the t-norm T and with quasi-arithmetic means. For non-strict continuous Archimedean t-norms they are completely characterised as generable by crisp equivalence relations. These fuzzy relations are used to define a method, called JADE, useful for feature selection and classification tasks. JADE is based on minimising the distance between two indistinguishability measures: the one given by weighting the attribute-values describing the domain objects and the other one given by the correct classification taken as an equivalence relation. The preliminary experiments we carried out with JADE are promising concerning the accuracy in solving classification tasks. We also report some issues of the method that could be improved in the future.

Keywords: T-generable indistinguishability operator; Similarity relation; Quasi-arithmetic mean; Feature selection; Classification

1. Introduction

Indistinguishability operators fuzzify the concept of equivalence relation. The flexibility in the selection of a specific t-norm to model their transitivity and the possibility of assessing degrees of relationship between the elements of a system make them very useful when the presence of uncertainty and inaccuracy requires a soft equivalence or equality.

One of the main issues on indistinguishability operators is their generation and representation. Every fuzzy subset μ of a universe X generates an indistinguishability operator E_μ on X in a very natural way [1]. In particular, for

* Corresponding author.
E-mail addresses: eva@iiia.csic.es (E. Armengol), dionis.boixader@upc.edu (D. Boixader), angel@iiia.csic.es, angel.garcia@upf.edu (À. García-Cerdaña), j.recasens@upc.edu (J. Recasens).

https://doi.org/10.1016/j.fss.2018.03.001
0165-0114/

a crisp set A of X, E_A is the equivalence relation associated to the partition of X generated by A (i.e., A and its complementary $\overline{A}$).

The important Representation Theorem [1] states that a fuzzy relation E on a set X is an indistinguishability operator if and only if there exists a family $(\mu_i)_{i\in I}$ of fuzzy subsets of X such that $E = \inf_{i\in I} E_{\mu_i}$. In this way an indistinguishability operator can be generated by a family of features or attributes expressed by the corresponding fuzzy subsets. Nevertheless, this way of generating indistinguishability operators presents a drawback: for a couple $x, y \in X$ only one fuzzy subset is used in the calculation of $E(x, y)$. Other methods have been proposed such as the use of weighted quasi-arithmetic means, OWA operators or calculating $T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$ [2–4].

Bezdek and Harris wrote the first paper dealing with the aggregation of T-indistinguishability operators and using the weighted arithmetic mean [5]. They study indistinguishability operators obtained as weighted arithmetic means of crisp equivalence relations. These authors also introduce a method for averaging crisp equivalence relations obtaining a fuzzy relation, although they do not characterise these T-indistinguishability operators.

Inspired by [5], we continue the study of the aggregation of indistinguishability operators in universes of finite cardinality for continuous Archimedean t-norms. First, we generalise the use of quasi-arithmetic means by showing that the weights do not need to sum up to one in order to obtain indistinguishability operators. Then we relate this result with powers with respect to the t-norm T as defined in [6–8]. T-generable indistinguishability operators E are those that can be obtained as $T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$, where $\mu_1, \mu_2, ..., \mu_m$ are fuzzy subsets of the universe. We define the set $M(X) = \{t^{[-1]}(\sum_{A\subseteq X} p_A \cdot t(E_A)) \mid p_A \geq 0\}$ of operators such that the t-norm T is non-strict continuous Archimedean, and t is one of its additive generators, and then we prove that E is T-generable if and only if it belongs to $M(X)$ (Proposition 2.24). Also T-indistinguishability operators obtained as quasi-arithmetic means of E_{μ_i} (where $i = 1, 2, ..., m$) are proved to be T-generable. In Section 2 we give some preliminaries explaining indistinguishable operators, fuzzy similarity relations and some interesting properties of them.

Indistinguishability (or Similarity) operators are the central issue of two subfields of Machine Learning: clustering and classification. The goal of clustering is to group objects by similarity. Behind classification, there is also the idea of similarity. However, here it is used during problem solving when the goal is to determine the class of a new unseen object. In Section 3 we explain how indistinguishability relations are used in machine learning mainly for solving classification tasks. In Section 4 the new problem solving method called JADE is introduced. We illustrate the explanation of the method with a running example. Section 5 shows the results of the experiments we carried out. These experiments are addressed to analyse the accuracy however, due to the complexity of JADE, we have led to explore some procedures to cluster the input data base. We also analyse how the partition of the data set influences the performance of the method. Finally, Section 6 is devoted to conclusions and future work.

2. T-generable indistinguishability operators

In many real situations the objects do not necessarily satisfy a property categorically, but rather satisfy it only at some level or degree (think for example of the property *to be rich*). In these cases, properties are fuzzy concepts and in particular we can not talk about completely equivalent objects, but a certain degree of similarity must be introduced. In this way, the equivalence becomes to a fuzzy concept and must be based on the concept of indistinguishability operator that is formalised in Definition 2.6.

First, let us recall some basic facts on continuous t-norms. In this section some already known results will be referred to suitable literature.

Definition 2.1. [7] A continuous t-norm is a map $T : [0, 1] \times [0, 1] \to [0, 1]$ such that for all $x, y, z \in [0, 1]$ satisfies

1. $T(T(x, y), z)) = T(x, T(y, z))$ (Associativity)
2. $T(x, y) = T(y, x)$ (Commutativity)
3. $T(1, x) = x$
4. T is a non-decreasing map
5. T is a continuous map

Commutativity can be derived from the other properties although the proof is not trivial.

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

Example 2.2. [7]

1. The minimum t-norm min defined by $\min(x, y)$ for all $x, y \in [0, 1]$.
2. The t-norm of Łukasiewicz defined by $T(x, y) = \max(0, x + y - 1)$.
3. The Product t-norm $T(x, y) = x \cdot y$.

This paper will only deal with continuous Archimedean t-norms. Throughout the paper we will use the following characterisation of these t-norms.

Theorem 2.3. *[7] A continuous t-norm T is Archimedean if and only if there exists a continuous and strictly decreasing function $t : [0, 1] \to [0, \infty)$ with $t(1) = 0$ such that*

$$T(x, y) = t^{[-1]}(t(x) + t(y))$$

where $t^{[-1]}$ is the pseudo inverse of t, defined by

$$t^{[-1]}(x) = \begin{cases} t^{-1}(x) & \text{if } x \in [0, t(0)] \\ 0 & \text{otherwise.} \end{cases}$$

T is strict if $t(0) = \infty$, and non-strict otherwise. The function t is called an additive generator of T and two generators of the same t-norm differ only by a positive multiplicative constant.

As it is clear from the definition of $t^{[-1]}$, $t^{[-1]}(x)$ can be replaced by $t^{-1}(x)$ whenever $x \in [0, t(0)]$.

Example 2.4. [7]

1. $t(x) = 1 - x$ is an additive generator of the t-norm of Łukasiewicz.
2. $t(x) = -\log(x)$ is an additive generator of the Product t-norm.

Definition 2.5. [7] Let T be a left continuous t-norm.

1. The residuation $\overrightarrow{T}$ of T is defined for all $x, y \in [0, 1]$ by
$$\overrightarrow{T}(x|y) = \max\{\alpha \in [0, 1] \mid T(\alpha, x) \leq y\}$$
2. The biresiduation $\overleftrightarrow{T}$ of T is defined for all $x, y \in [0, 1]$ by
$$\overleftrightarrow{T}(x, y) = \min(\overrightarrow{T}(x|y), \overrightarrow{T}(y|x))$$

Definition 2.6. [9,10] Let X be a universe and T a t-norm. A T-indistinguishability operator or similarity relation E on X is a fuzzy relation $E : X \times X \to [0, 1]$ on X, satisfying for all $x, y, z \in X$ the following properties:

1. $E(x, x) = 1$ (Reflexivity)
2. $E(x, y) = E(y, x)$ (Symmetry)
3. $T(E(x, y), E(y, z)) \leq E(x, z)$ (T-Transitivity)

$E(x, y)$ is interpreted as the degree of similarity, equivalence, or indistinguishability between x and y.

A subset A of a universe X defines a partition of X into two classes in a straightforward way: A and its complementary set $\overline{A}$. This partition has associated an equivalence relation. In a similar way, we will see in Proposition 2.7 that a fuzzy subset μ of X generates a similarity relation E_μ in a natural way. Hence, the attributes of the objects of X, considered as fuzzy subsets, generate similarity relations (indistinguishability operators) on X.

Proposition 2.7. *[1] Let μ be a fuzzy subset of X. The fuzzy relation on X defined for all $x, y \in X$ by $E_\mu(x, y) = \overleftrightarrow{T}(\mu(x), \mu(y))$ is a T-indistinguishability operator.*

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

A particular case of this proposition is the following corollary:

Corollary 2.8. *[1] If T is a continuous Archimedean t-norm and t an additive generator of T, then $E_\mu(x, y) = t^{-1}(|t(\mu(x)) - t(\mu(y))|)$.*

In particular,

- *If Ł is the Łukasiewicz t-norm, then $E_\mu(x, y) = 1 - |\mu(x) - \mu(y)|$.*
- *If P is the Product t-norm, then $E_\mu(x, y) = \min(\frac{\mu(x)}{\mu(y)}, \frac{\mu(y)}{\mu(x)})$.*

Lemma 2.9. *If $A \subseteq X$ is a crisp subset of X, then E_A is the crisp equivalence relation generated by the partition $\{A, \overline{A}\}$ of X generated by A*

$$E_A(x, y) = \begin{cases} 1 & \text{if } x, y \in A \text{ or } x, y \in \overline{A} \\ 0 & \text{otherwise} \end{cases}$$

for all $x, y \in X$.

Proof. Trivial. □

Lemma 2.10. *Let $A \subseteq X$ be a subset of X and $\overline{A}$ its complementary. Then $E_A = E_{\overline{A}}$.*

Proof. Trivial. □

Definition 2.11. [2] Let T be a continuous Archimedean t-norm with additive generator t. For weights $p_1, p_2, ..., p_k \in [0, 1]$ such that $\sum_{i=1}^{k} p_i = 1$ we can associate to T the quasi-arithmetic mean $m_t^{p_1,p_2,...,p_k}$ defined for all $x_1, x_2, ..., x_k \in [0, 1]$ as follows

$$m_t^{p_1,p_2,...,p_k}(x_1, x_2, ..., x_k) = t^{-1}(\sum_{i=1}^{k} p_i t(x_i)).$$

Example 2.12. [2]

- If T is the t-norm of Łukasiewicz, then m_t is the weighted arithmetic mean.
- If T is the Product t-norm, then m_t is the weighted geometric mean.

It is known [2,3] that the weighted quasi-arithmetic mean $m_t^{p_1,p_2,...,p_k}$ (where t is an additive generator of T) of a finite family of T-indistinguishability operators on a set X is also a T-indistinguishability operator on X. Next proposition generalises this result by imposing only that the numbers p_i have to be non-negative.

Proposition 2.13. *Let T be a continuous Archimedean t-norm, t an additive generator of T, $E_1, E_2, ..., E_k$ T-indistinguishability operators on a set X, and $p_1, p_2, ..., p_k$ non-negative real numbers. Then the fuzzy relation E on X defined for all $x, y \in X$ as*

$$E(x, y) = t^{[-1]}(p_1 \cdot t(E_1(x, y)) + p_2 \cdot t(E_2(x, y)) + ... + p_k \cdot t(E_k(x, y)))$$

is a T-indistinguishability operator on X.

The fuzzy relation E can also be denoted by $m_t^{p_1,p_2,...,p_k}(E_1, E_2, ..., E_k)$.

When the sum of the numbers p_i is less than or equal to one, the last proposition can be rewritten in the following way in the case.

Proposition 2.14. *Let T be a continuous Archimedean t-norm, t an additive generator of T, $E_1, E_2, ..., E_k$ T-indistinguishability operators on a set X, and $p_1, p_2, ..., p_k$ non-negative real numbers with $\sum_{i=1}^{k} p_i \leq 1$. Then the*

T-indistinguishability operator E on X defined by $m_t^{p_1,p_2,...,p_k}(E_1, E_2, ..., E_k)$ is the weighted quasi-arithmetic mean $m_t^{p_1,p_2,...,p_k,1-\sum_{i=1}^{k} p_i}(E_1, E_2, ..., E_k, \mathbf{1})$ where $\mathbf{1}$ is the universal T-indistinguishability operator $\mathbf{1}(x, y) = 1$ for all $x, y \in X$.

Proof. Trivial, since $t(1) = 0$. □

In [6–8], the power of an element $x \in [0, 1]$ with respect to a positive real number p and a given t-norm T is defined generalising the power $x^{(n)} = T(\overbrace{x, x, ..., x}^{n \text{ times}})$ $n \in \mathbb{N}$ to any $p \geq 0$. This generalization has been fruitful in different fields as can be seen, for instance, in [11–13]. For continuous Archimedean t-norms the following result is known.

Proposition 2.15. *[6] Let T be a continuous Archimedean t-norm, t an additive generator of T and $p \geq 0$. Then,*

- $x^{(p)} = t^{[-1]}(p \cdot t(x))$ *for all* $x \in [0, 1]$.
- $E^{(p)}$ *is a T-indistinguishability operator on X if E is.*

Thanks to Proposition 2.14, $x^{(p)}$ can be interpreted as an average between x and 1 if $0 \leq p \leq 1$.

Proposition 2.16. *Let T be a continuous Archimedean t-norm, t an additive generator of T and $p \in [0, 1]$. Then,*

- $x^{(p)} = m_t^{p,1-p}(x, 1)$ *for all* $x \in [0, 1]$.
- $E^{(p)} = m_t^{p,1-p}(E, \mathbf{1})$ *for all T-indistinguishability operators E on X.*

The next lemma shows that the power $E^{(p)}$ of a T-indistinguishability operator generated by a fuzzy subset μ can also be generated by the fuzzy subset $\mu^{(p)}$.

Lemma 2.17. *Let T be a continuous Archimedean t-norm, t an additive generator of T, X a set, μ a fuzzy subset of X, and $p \in [0, 1]$. Then $(E_\mu)^{(p)} = E_{\mu^{(p)}}$.*

Proof. For $x, y \in X$,

$$(E_\mu)^{(p)}(x, y) = t^{-1}(p \cdot |t(\mu(x)) - t(\mu(y))|) = t^{-1}(|p \cdot t(\mu(x)) - p \cdot t(\mu(y))|)$$

Since $\mu^{(p)} = t^{-1}(p \cdot t(\mu))$, the last member is equal to $E_{\mu^{(p)}}$. □

We next define the concept of T-generability for a T-indistinguishability operator.

Definition 2.18. Let T be a t-norm, X a set and E a T-indistinguishability operator on X. Then E is T-generable if there exists a finite family $\mu_1, \mu_2, ..., \mu_m$ of fuzzy subsets of X such that $E = T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$.

Proposition 2.19. *Let T be a continuous Archimedean t-norm, t an additive generator of T, X a finite set, $\mu_1, \mu_2, ..., \mu_m$ fuzzy subsets of X, and $p_1, p_2, ..., p_m \in [0, 1]$. Then the T-indistinguishability operator*

$$E = m_t^{p_1,p_2,...,p_m}(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$$

on X is T-generable.

Proof.

$$\begin{aligned} E &= m_t^{p_1,p_2,...,p_m}(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m}) \\ &= t^{-1}(p_1 \cdot t(E_{\mu_1}) + p_2 \cdot t(E_{\mu_2}) + ... + p_m \cdot t(E_{\mu_m})), \end{aligned}$$

$\mu_i^{(p)} = t^{-1}(p_i \cdot t(\mu_i))$ $i = 1, 2, ..., m$ and by Lemma 2.17, the equation above equals

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

$$t^{-1}(t(E_{\mu_1^{(p)}}) + t(E_{\mu_2^{(p)}}) + ... + t(E_{\mu_m^{(p)}})) =$$
$$= T(E_{\mu_1^{(p)}}, E_{\mu_2^{(p)}}, ..., E_{\mu_m^{(p)}}). \quad \square$$

In particular, quasi-arithmetic means of T-indistinguishability operators generated by a fuzzy subset are T-generable.

The crisp equivalence relations E_A generated by a crisp subset A of X will play a central role for characterizing T-generable indistinguishability operators.

Definition 2.20. Let T be a continuous Archimedean t-norm, t an additive generator of T, and X a finite set. The set $M(X)$ of T-indistinguishability operators on X is defined by

$$M(X) = \{t^{[-1]}(\sum_{A \subseteq X} p_A \cdot t(E_A)) \mid p_A \geq 0\}.$$

If the t-norm is strict, then $M(X)$ is the set of all crisp equivalence relations on X (Proposition 2.21). Most interesting case is when T is non-strict since in such situation $M(X)$ coincides with the set of T-generable indistinguishability operators (Proposition 2.24).

Proposition 2.21. *Let T be a strict continuous Archimedean t-norm, t an additive generator of T, and X a finite set. Then $M(X)$ is the set of crisp equivalence relations on X.*

Proof. a) If $E \in M(X)$, then $E = t^{[-1]}(\sum_{A \subseteq X} p_A \cdot t(E_A))$. For $x, y \in X$ and $A \subseteq X$, $E_A(x, y)$ equals either 0 or 1. So $t(E_A(x, y))$ is either ∞ or 0 and so is $\sum_{A \subseteq X} p_A \cdot t(E_A(x, y))$.

b) Let E be a crisp equivalence relation and $\{A_1, A_2, ..., A_m\}$ the associated partition of X in its equivalence classes. Then it is immediate to see that $E = t^{[-1]}(\sum_{i=1}^{m} p_i \cdot t(E_{A_i}))$, for any $p_1, p_2, ..., p_m > 0$. $\square$

Proposition 2.22. *Let T be a non-strict continuous Archimedean t-norm, t an additive generator of T, and X a finite set. Then the crisp equivalence relations on X are in $M(X)$.*

Proof. The same as in Proposition 2.21b), but considering $p_1, p_2, ..., p_m \geq 1$. $\square$

Lemma 2.23. *Let T be a non-strict Archimedean t-norm, t an additive generator of T, and μ a fuzzy subset of the finite set $X = \{x_1, x_2, ..., x_n\}$. Then $E_\mu \in M(X)$.*

Proof. We can assume without loss of generality that $\mu(x_i) \leq \mu(x_j)$ if $i < j$, therefore

$$E_\mu = t^{-1}(\frac{1}{t(0)} \sum_{i=1}^{n-1} (t(\mu(x_{i+1})) - t(\mu(x_i)) \cdot t(E_{\{x_1, x_2, ..., x_i\}})).$$

Taking into account both $t(1) = 0$ and

$$E_{\{x_1, x_2, ..., x_i\}}(x_j, x_{j+k}) = \begin{cases} 1 & \text{if } j + k \leq i \text{ or } j > i \\ 0 & \text{if } j \leq i < j + k \end{cases}$$

we have that

$$t^{-1}(\frac{1}{t(0)} \sum_{i=1}^{n-1} (t(\mu(x_{i+1})) - t(\mu(x_i)) \cdot t(E_{\{x_1, ..., x_i\}}(x_j, x_{j+k}))$$
$$= t^{-1}(\frac{1}{t(0)} \sum_{i=j}^{j+k-1} (t(\mu(x_{i+1})) - t(\mu(x_i)) \cdot t(0)) = t^{-1}(t(x_{j+k}) - t(x_j))$$
$$= t^{-1}(|t(x_{j+k}) - t(x_j)|) = E_\mu(x_j, x_{j+k}). \quad \square$$

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

In the non-strict case, the next proposition characterises the T-generable indistinguishability operators as the ones in $M(X)$.

Proposition 2.24. *Let T be a non-strict continuous Archimedean t-norm, t an additive generator of T, $X = \{x_1, x_2, ..., x_n\}$ a finite set, and E a T-indistinguishability operator on X. Then E is T-generable if and only if $E \in M(X)$.*

Proof.

$\Leftarrow$) If $E \in M(X)$, then there exist some subsets $A_1, A_2, ..., A_m$ of X and $p_1, p_2, ..., p_m > 0$ such that $E = t^{[-1]}(\sum_{i=1}^{m} p_i \cdot t(E_{A_i}))$.
For each $i = 1, 2, ..., m$ we can define the fuzzy subset μ_i of X:

$$\mu_i(x_j) = \begin{cases} 1 & \text{if } x_j \in A_i \\ t^{[-1]}(t(0) \cdot p_i) & \text{otherwise.} \end{cases}$$

Therefore $E = T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$.

$\Rightarrow$) Suppose $E = T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m})$. From Lemma 2.23, for $l = 1, 2, ..., m$

$$E_{\mu_l} = t^{-1}(\frac{1}{t(0)} \sum_{i=1}^{k_l} p_{l_i} E_{A_{l_i}})\ k_l \in \mathbb{N}\ p_{l_i} \geq 0\ A_{l_i} \subseteq X,$$

$$E = T(E_{\mu_1}, E_{\mu_2}, ..., E_{\mu_m}) = t^{[-1]}(t(E_{\mu_1}) + t(E_{\mu_2}) + ... + t(E_{\mu_m}))$$

$$= t^{[-1]}(t(t^{-1}(\frac{1}{t(0)} \sum_{i=1}^{k_1} p_{1_i} E_{A_{1_i}})) + ... + t(t^{-1}(\frac{1}{t(0)} \sum_{i=1}^{k_m} p_{m_i} E_{A_{m_i}})))$$

$$= t^{[-1]}(\frac{1}{t(0)} \sum_{i=1}^{k_1} p_{1_i} E_{A_{1_i}} + \cdots + \frac{1}{t(0)} \sum_{i=1}^{k_m} p_{m_i} E_{A_{m_i}}). \quad \square$$

Corollary 2.25. *Let T be a non-strict continuous Archimedean t-norm and X a finite set. Then the crisp equivalence relations on X are T-generable.*

Proof. It is a consequence of Proposition 2.22. $\square$

Definition 2.26. Let T be a t-norm and E a T-generable indistinguishability operator on a set X. The minimum number of fuzzy subsets of X needed to T-generate E is called its T-dimension ($dim_T(E)$).

From the Proposition 2.24 we obtain the following corollary.

Corollary 2.27. *Let T be a non-strict continuous Archimedean t-norm, t an additive generator of T, X a finite set, and E a T-generable indistinguishability operator on X. If $E = t^{[-1]}(\sum_{i=1}^{m} p_i \cdot t(E_{A_i}))$, then $dim_T(E) \leq m$.*

The next result provides a (very rough) upper bound to the T-dimension of a T-generable indistinguishability operator.

Corollary 2.28. *Let T be a non-strict continuous Archimedean t-norm, t an additive generator of T, X a finite set of cardinality n, and E a T-generable indistinguishability operator on X. Then, $dim_T(E) \leq 2^{n-1}$.*

Proof. The maximum number of subsets of X involved in the representation $E = t^{[-1]}(\sum_{i=1}^{m} p_i \cdot t(E_{A_i}))$ with $p_i \neq 0$ for $i = 1, 2, ..., m$ is 2^n. Since $E_A = E_{\overline{A}}$ (Lemma 2.10), we get the upper bound. $\square$

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

3. Indistinguishability operators in Machine Learning

Machine Learning algorithms can be classified in two families according to the nature of the input examples: *supervised* and *unsupervised*. Supervised learning methods handle labelled input examples. For instance, let us suppose we have a set of animals $\mathcal{A}$ that according to their morphological description can be classified as *bird, reptile* or *fish*. The labels indicate the membership of an animal to one of these groups. Let x be a new unseen animal we want to classify. The process for its classification is to assess the similarity of x to each one of the known animals of the set $\mathcal{A}$ and to determine the class of x according to these similarities. This is the idea of the k-NN algorithm [14], the basic algorithm of *Case-based Reasoning* (CBR) [15]. The k-NN algorithm classifies an unseen object based on the classes of the k most similar objects. Domain objects are commonly described as attribute-value pairs where such values can be both categorical or numerical.

The global similarity between two domain objects is assessed by measuring the local similarity of each one of the attributes and then, by aggregating these local similarities. Thus, given two domain objects $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ described by n attributes (where x_i and y_i are the values of the ith attribute of X and Y respectively), the global similarity between X and Y is assessed as follows:

$$\mathrm{Sim}(X, Y) = @_{i=1}^{n}\mathrm{sim}(x_i, y_i),$$

where @ is an aggregation function (for instance, the mean, the weighted mean, the OWA operator, etc), and $\mathrm{sim}(x_i, y_i)$ is the similarity between the values of the attribute ith of both objects. For instance, using the mean to aggregate the local similarities, we have:

$$\mathrm{Sim}(X, Y) = \frac{1}{n}\sum_{i=1}^{n}\mathrm{sim}(x_i, y_i). \tag{1}$$

When the values of the attributes are numerical, sim is usually the dual of some normalised distance measure, for instance the Euclidean distance, the Minkowski distance or the Mahalanobis distance. Otherwise, when the values are categorical, the usual measure is the following:

$$\mathrm{sim}(x_i, y_i) = \begin{cases} 1, & \text{if } x_i = y_i \\ 0, & \text{otherwise.} \end{cases}$$

Notice that in Eq. (1) all the attributes describing the domain objects have the same importance (weight). However, it should be possible to consider that some attributes are more important than others and to assess a different weight to them. In such situation the Sim function should be

$$Sim(X, Y) = \sum_{i=1}^{n} p_i \cdot sim(x_i, y_i), \text{ with } \sum_{i=1}^{n} p_i = 1.$$

A different approach of supervised learning methods is the one taken by the *inductive learning methods* [16], where the goal is to construct a domain model that will be further used for classifying unseen objects. Such model is composed of discriminant descriptions for each class. Each description of a class C_i has the attributes considered as the most relevant for classifying an object as belonging to C_i. The key issue here is to determine such relevant attributes and several common measures are used to this purpose: the Gini's index, the Quilan's Information Gain, the Variance Reduction, etc (see [16]). Most of these measures assess the homogeneity resulting from selecting a particular attribute. The goal is to separate as much as possible the input objects according to the classes. The final descriptions have to be satisfied by objects of only one of the classes.

Unsupervised learning methods are used when input examples have not class labels. Common methods here are *clustering* methods. The goal of clustering methods is to group the examples by similarity. These groups are called *clusters* and each cluster satisfies that all its elements are more similar between them than to elements of other clusters. As in inductive learning methods, here is also key to determine which attributes are relevant to form the clusters.

The assessment of weights to attributes in both, inductive learning methods and clustering, is closely related with feature selection methods. Feature selection and feature extraction methods' goal is to detect and eliminate redundant information and take only those features considered relevant for the task at hand. The selection of the appropriate features greatly influences the quality of the data used by the algorithms and, thus, the goodness of the final result.

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

Most of feature extraction and feature selection methods are based on similarities. One of the earliest feature selection methods is RELIEF [17], used in binary classification, which is an iterative algorithm that adjust the weight of an attribute using the formula:

$$W_i' = W_i - (x_i - \text{nearHit}_i)^2 + (x_i - \text{nearMiss}_i)^2,$$

where nearHit_i is the closest instance to x_i belonging to the same class and nearMiss_i is the closest instance to x_i belonging to a different class. Features are selected if their relevance is greater than a given threshold.

A different approach is the one taken in QPFS [18] where the feature selection problem is reformulated as an optimisation problem. The QPFS method works with an objective function that has two terms: one that is quadratic and another one that is linear. The quadratic term captures the dependence between each pair of variables whereas the linear term captures the relationship between each feature and the class label. Both terms are weighted by a parameter α that depends on the domain and that regulates what is preferred in such domain either the quadratic term or the linear term. The weights of the features are calculated using quadratic programming. In [18] the QPFS method is explained and also experimental results are reported. A similar approach is the one introduced by Zhang et al., [19] where the weights are also computed using quadratic programming. In the next section we propose a new approach, namely JADE, that resembles to both QPFS and the method in [19] in the feature selection task since we also reformulate it as an optimisation method.

4. The JADE method

In this section we introduce JADE, a method useful for feature selection and classification, where the problem to assess the weights of the attributes has been reformulated as an optimisation problem like the methods in [18] and [19]. Conceptually, the idea is to minimise the distance between two indistinguishability relations: the one that gives the correct classification of the known examples and the other one that is a linear combination of the indistinguishable operators generated by the attributes describing the examples. Such distance is calculated using the Euclidean distance, so the function to be minimised is a quadratic one. As we prove in the experiments, JADE can be used for both feature selection and also as a classifier. Let us explain JADE in detail.

Let X be a set of labelled domain objects described by a set of attributes $A = \{a_1, a_2, \ldots, a_n\}$, where the attributes a_i are considered fuzzy subsets of X; and E_{a_i} the Ł-indistinguishability operator generated by a_i. Each $x_i \in X$ belongs to one solution class $\{C_1, C_2, \ldots, C_k\}$, i.e., there are k classes where an unseen domain object could be classified. This is also a difference with approaches as QPFS or [19] since they work only on domains with two solution classes (although some further modifications allow to deal with more than two classes). On the set X we can induce two kinds of partitions:

- The *correct partition* that is the one that separates the objects in X according the solution classes $C = \{C_1 \ldots C_k\}$.
- The partitions induced by each attribute in A. Given an attribute $a_j \in A$, the objects in X can be separated according to the value that they hold in the attribute a_j.

In terms of similarity relations, the correct partition can be seen as an equivalence relation R on X, and each partition induced by an attribute a_j can be seen as a local similarity relation E_{a_j} on the set X. Both R and each E_{a_j} can be represented as matrices such that:

- R is a $m \times m$ matrix (m being the number of objects in X) where each element r_{hl} is equal to 1 if the objects x_h and x_l belong to the same solution class and 0 otherwise.
- E_{a_j} is a $m \times m$ matrix where each element e_{hl} is the similarity that the objects x_h and x_l with respect to the attribute a_j.

Notice that the global similarity between two objects can be assessed by the weighted mean $E = \sum_{i=1}^{n} E_{a_i} p_i$ of these local similarity relations.

Based on this, JADE considers the objective function as a distance function that measures how different (or similar) are two similarity relations: the one given by the global similarity of the attribute-value pairs describing the domain objects (E) and the one given by the correct classification taken as an equivalence relation (R). The goal of JADE is

	a_1	a_2	a_3	Class
x	0.3	0.2	0.8	C
y	0.2	0.4	0.7	D
z	0.5	0.6	0.2	C

$R =$

	x	y	z
x	1	0	1
y	0	1	0
z	1	0	1

Fig. 1. Running example. The objects x, y and z are described by three attributes a_1, a_2 and a_3, and there are two solution classes: C and D. At the right part, the matrix R representing the class equivalence relation among the domain objects.

to assess the weights p_i for which E is closest to R. The weights p_i associated to a similarity relation E_{a_i} will denote the importance of the feature a_i in the process of classifying the examples of X.

Let $p_1, p_2, ..., p_n \geq 0$ (with $\sum_{i=1}^{n} p_i = 1$) be the n weights associated to the n attributes describing the objects in X. The relation

$$E = p_1 \cdot E_{a_1} + p_2 \cdot E_{a_2} + ... + p_n \cdot E_{a_n} \tag{2}$$

where the attributes a_i are considered as fuzzy subsets of X, and E_{a_i} is an Ł-indistinguishability operator on X (Proposition 2.13). The Euclidean distance d between the relations E and R is defined in the usual way as follows:

$$d(E, R) = \sqrt{\sum_{i,j=1..k} (E(x_i, x_j) - R(x_i, x_j))^2}, \tag{3}$$

where R is the equivalence relation associated to the correct partition and E is the Ł-indistinguishability operator in Eq. (2). The key issue in JADE is to minimise such distance, so we have to find the weights that minimise it. Therefore, the function in Eq. (3) is our objective function, the one we want to minimise using quadratic programming. In other words, our task is

$$\begin{aligned} &\text{minimise} && d(E, R) \\ &\text{subject to} && p_1, p_2, ..., p_n \geq 0 \\ & && \sum_{i=1}^{n} p_i = 1. \end{aligned}$$

The attributes generating similarity relations with higher weights help more to the resemblance of E to R. The weights of the attributes give an idea of which of them are the more relevant to describe a class (notice that the weight of some attributes could be zero). Also, the weights can be used to directly classify unseen objects since, following the idea of the k-NN algorithm, a new example y will be classified in the class of the example to which is more similar.

Even in the case in which all attributes are categorical – they generate crisp equivalence relations – the obtained relation E from them would be fuzzy containing all the information of these equivalence relations and weighted by adequate weights. This would not be possible if we were trying to find a crisp relation E.

In the next sections we will explain with a running example how JADE works for assessing the weights to the attributes and also how can be used for classifying unseen objects.

4.1. Running example

Let x, y, z be domain objects described by three attributes a_1, a_2 and a_3 and belonging to one of the two solution classes C and D (Fig. 1). The partition induced by the classification of the objects is the equivalence relation R also shown in Fig. 1. Each component r_{ij} of R is equal to 1 if the corresponding objects belong to the same solution class and 0 otherwise. Thus, for instance, $r_{13} = 1$ because both x and z belong to the same class, and $r_{12} = 0$ because x and y belong to different classes.

The continuous attributes a_1, a_2 and a_3 generate Ł-indistinguishability operators E_{a_1}, E_{a_2} and E_{a_3} (as shown in Corollary 2.8) that are matrices whose elements can be calculated using the biresiduation of the Łukasiewicz t-norm defined as $\overleftrightarrow{T}(x, y) = 1 - |x - y|$.

Thus, for instance, the elements of E_{a_1} have been calculated as follows:

$$E_{a_1}(x.a_1, y.a_1) = 1 - |x.a_1 - y.a_1| = 1 - |0.3 - 0.2| = 0.9$$

$$E_{a_1}(x.a_1, z.a_1) = 1 - |x.a_1 - z.a_1| = 1 - |0.3 - 0.5| = 0.8$$
$$E_{a_1}(y.a_1, z.a_1) = 1 - |y.a_1 - z.a_1| = 1 - |0.2 - 0.5| = 0.7,$$

where $x.a_i$ stands for the value of a_i of the object x (respectively, of the objects y and z) The elements of E_{a_2} and E_{a_3} have been calculated in a similar way. Finally, we obtain the following matrices:

$$E_{a_1} = \begin{pmatrix} 1 & 0.9 & 0.8 \\ 0.9 & 1 & 0.7 \\ 0.8 & 0.7 & 1 \end{pmatrix}, \quad E_{a_2} = \begin{pmatrix} 1 & 0.8 & 0.6 \\ 0.8 & 1 & 0.8 \\ 0.6 & 0.8 & 1 \end{pmatrix},$$
$$E_{a_3} = \begin{pmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.5 \\ 0.4 & 0.5 & 1 \end{pmatrix}.$$

In this example, the global similarity among the objects x, y and z, is given by the expression (Eq. (2)): $E = p_1 \cdot E_{a_1} + p_2 \cdot E_{a_2} + p_3 \cdot E_{a_3}$. Therefore, the goal is to minimise the objective function (Eq. (3)) taking into account that the weights must be numbers between 0 and 1, the problem turns into a quadratic programming one [20] since we have to optimise (minimise) the function $d(E, R)$ in the feasible region.

Therefore, in our running example the goal is the following one:

minimise:

$$d(E,R)^2 = 2\big((E(x,y) - R(x,y))^2 + (E(x,z) - R(x,z))^2 + (E(y,z) - R(y,z))^2\big)$$
$$= 2\big((0.9p_1 + 0.8p_2 + 0.9p_3 - 0)^2 + (0.8p_1 + 0.6p_2 + 0.4p_3 - 1)^2 + (0.7p_1 + 0.8p_2 + 0.5p_3 - 0)^2\big),$$

under the conditions:

$$0 \le p_1 \le 1,\ 0 \le p_2 \le 1,\ 0 \le p_2 \le 1, \text{ and } p_1 + p_2 + p_3 = 1.$$

The solution for our running example is the set of weights: $p_1 = 0.7$, $p_2 = 0$ and $p_3 = 1 - (p_1 + p_2) = 0.3$. Therefore, the importance of the attributes is given by the following order: a_1, a_3 and finally a_2.

4.2. How to classify a new object?

In the previous section, we have seen how to asses the weight of each attribute. Now, we can take benefit of these weights for classifying unseen objects. Let us suppose a new object v taking the values 0.6, 0.5, 0.3 on the features a_1, a_2 and a_3 respectively. We want to classify v into either C or D.

The first step is to assess the similarity of each one of the attributes of v to each one of the attributes of the known examples x, y and z. Thus, for E_{a_1} we have:

$$E_{a_1}(x.a_1, v.a_1) = 1 - |x.a_1 - v.a_1| = 1 - |0.3 - 0.6| = 0.7$$
$$E_{a_1}(y.a_1, v.a_1) = 1 - |x.a_1 - v.a_1| = 1 - |0.2 - 0.6| = 0.6$$
$$E_{a_1}(z.a_1, v.a_1) = 1 - |x.a_1 - v.a_1| = 1 - |0.5 - 0.6| = 0.9$$

Therefore, the three Ł-indistinguishability operators E_{a_1}, E_{a_2} and E_{a_3} of the previous subsection have to be extended with a fourth file (and column, since the matrix is symmetric) with the similarities of the new object v. These fourth columns are, respectively:

$$E_{v,a_1} = (0.7, 0.6, 0.9, 1),\ E_{v,a_2} = (0.7, 0.9, 0.9, 1),\ E_{v,a_3} = (0.5, 0.6, 0.9, 1).$$

Now, aggregating E_{a_1}, E_{a_2} and E_{a_3} with the weights p_1, p_2 and p_3 found in the previous section we obtain the similarity relation E:

$$E = 0.7E_{a_1} + 0E_{a_2} + 0.3E_{a_3} = \begin{pmatrix} 1 & 0.90 & 0.68 & 0.64 \\ 0.90 & 1 & 0.64 & 0.60 \\ 0.68 & 0.64 & 1 & 0.90 \\ 0.64 & 0.60 & 0.90 & 1 \end{pmatrix}$$

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

Looking at the fourth row (or column) we see that the similarity degrees of v to the examples x, y, z are 0.64, 0.6 and 0.9 respectively. Because the most similar object to v is z, v will be classified as belonging to the class of z, i.e., to C.

5. JADE in practise

We have experimented with the JADE method on two well-known data sets from the UCI Machine Learning Repository [21]: *Iris* and *Zoology*. The *Iris* data set is composed of 150 objects belonging to three solution classes. Each object is described by four continuous attributes. The *Zoology* data set is composed of 100 objects belonging to 7 solution classes. Each object is described by 17 categorical attributes.

Our plan was to use the well known 10-fold cross-validation method to assess the predictivity of JADE. However, JADE has a high complexity, since it has to handle n matrices (where n is the number of attributes of the domain objects) having each one a dimension of $N \times N$ (where N is the number of objects of the dataset). For this reason, we have to pre-process in some way the input data in order to reduce the number of objects to be handled by JADE. In the following sections we explain the pre-process we performed on the input data and also the result of the experiments.

5.1. Pre-processing the input data

The goal of pre-processing the data is to reduce the complexity of JADE by reducing the number of input examples that it has to handle. A way to perform such reduction is to partition the set of input examples in smaller subsets and then use JADE on each one of these subsets. The criteria we have taken to form the subsets is by similarity. By fixing a threshold of similarity h, a subset is formed by all the input objects x_i and x_j such that $sim(x_i, x_j) \geq h$. Assuming that the attributes of the domain objects are continuous, they have been normalised, we used the following algorithm:

1. To generate a matrix S, where each element $s_{ij} \in S$ is the similarity ($1 - Euclidean\ distance$) of the input objects x_i and x_j.
2. To find the two objects with greatest similarity and take one of them, say p.
3. To construct a cluster C_p with all the objects such that $sim(p, x_i) \geq h$ and where p is its prototype.
4. To use JADE to find the coefficients that characterise the cluster C_p.
5. To consider all the objects in C_p as used and then select a new object q from the remaining ones such that $sim(p, q) < h$ and with maximum similarity to p.
6. Repeat all the steps from 3 to 5 until all the objects in the data set have been included in some cluster.

The result of this process is a set of k clusters, each one characterised by a set of coefficients that minimise the Eq. (3).

5.2. Classification of unseen objects

Now we have k clusters where the objects have been grouped by similarity, however each cluster can include objects of several classes. We also have the coefficients that give the importance of each attribute inside each cluster. How can we classify unseen objects? We have several options and we have experimented with all of them. Let x_i be the object to be classified.

- **Version 1 (V1)**. Each cluster has a prototype p_j that is the element from which the cluster has been constructed, i.e., all other objects x_m in the cluster have similarity $sim(p_j, x_m) \geq h$. We use the coefficients of the cluster whose prototype has maximum similarity to the new object x_i. Notice that they can exist several prototypes having the same similarity to x_i. In such situation we use the coefficients of all these clusters to classify x_i and then the majority rule to propose a final classification for x_i.
- **Version 2 (V2)**. For each cluster we can count how many objects x_j have $sim(x_j, x_i) \geq h$, and use the coefficients of the cluster having the highest number of similar objects. If several clusters have the same number of similar objects, we use all of them and then the majority rule to propose the final classification. If there are not objects with similarity equal or higher than h, this version does not propose a classification for x_i.

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

Table 1
Description of the UCI datasets used in the experiments: type of the attributes, number of classes, threshold (h) used to form the clusters, size of both the training and the test sets used in the experiments, and the mean accuracy for each version.

Data set	Type	Classes	h	# train	# test	Accuracy (%)		
						$V1$	$V2$	$V3$
Iris	Cont.	3	0.9	100	50	96.6	87.8	96.2
Zoology	Categ.	7	0.8	90	10	92.0	94.0	95.0

Table 2
Comparison of accuracies (in %) between the methods JADE with $V3$, the decision-tree based algorithm J48, Random Forest (RF), Naive Bayes (NB), Multilayer Perceptron (MP), and Instance-based algorithm (IB).

Data set	JADE	J48	RF	NB	MP	IB
Zoology	95	96	–	–	–	–
Iris	96.2	96	95.33	96	95.33	97.33

- **Version 3 (V3)**. To assess the global similarity of each cluster to the new object and use the cluster having the maximum similarity. The global similarity of a cluster C_k is assessed as the mean of all the similarities between the new object x_i and each one of the objects in C_k. If several clusters have the same global similarity, we use all of them and then the majority rule to propose the classification for x_i.

5.3. Accuracy

We performed experiments with several similarity thresholds h. Notice that for thresholds near to 1, the elements of a cluster will be very similar and possibly they belong to the same solution class. However, the number of clusters could be high and with few elements. Conversely, with low thresholds, clusters include many elements that possibly belong to different classes. We have also seen that for objects represented with attributes with continuous values, thresholds have to be higher than the ones used when attributes have categorical values. We carried out experiments with values of h from 0.7 to 0.95. Table 1 shows the highest accuracy obtained and the threshold h associated to it. These results have been obtained after 10-fold-cross validation.

In these experiments we have seen that continuous and categorical attributes have slightly different requirements. Thus, the best threshold of similarity h is higher in *Iris* (continuous) than in *Zoology* (categorical). This is an expected result because differences between objects are more fine grained when the attributes are continuous. Also we have seen that for *Iris*, best versions are $V1$ and $V3$, i.e., the one based on prototypes and the one based on global similarities, whereas on *Zoology* the version $V1$ is the worst. Again, we think that this is due to the type of the attributes (continuous attributes vs categoric ones). This issue should be further analysed by using JADE on other categorical and continuous domains.

We compared the JADE accuracy with the one of several classification methods provided by the Weka application [22] on *Iris*. The results obtained with JADE on *Zoology* were compared only with the results using the J48 algorithm because of this domain has categorical attributes. Table 2 shows the accuracy results of all these methods after one trial of 10-fold cross-validation. Notice that JADE has a performance similar to the one exhibited by the tree-based classifier J48 in both domains with $V3$. Also, the JADE performance is lower than the one exhibited by an Instance-based learning algorithm (with several values of k), comparable to the one of J48 and Naive Bayes, and higher than the accuracy exhibited by both Random Forest and Multilayer Perceptron.

5.4. Analysing the clusters

As we have seen in the previous section, the accuracy of JADE is comparable to the one of the standard method J48. However, to reduce the complexity we have to cluster the input data base and each set of this partition has been used as input to JADE, resulting in a set of coefficients for each cluster. We have used a k-NN-like method to do the partition,

i.e., each cluster is well-formed since it contains similar objects. The question now is, how these well-formed clusters influence the accuracy of JADE? Will the accuracy of JADE be higher if the input data set is randomly clustered?

We have the following two hypotheses:

- **H1**. When the clusters are well-formed, i.e., its elements are very similar, the coefficients characterising the clusters are more accurate and therefore, the accuracy of JADE will be higher.
- **H2**. If the clusters contain elements of different classes, the coefficients could discriminate better among the classes.

Notice also that **H1** has some shortcoming when classifying unseen objects. Let us suppose a data set where its objects belong to two solution classes: $C1$ and $C2$. Let us consider an extreme case where the data set has been partitioned in two correct clusters: S_1 that contains only elements of $C1$; and S_2 that contains only elements of $C2$. Let us suppose now that we are using the version $V1$ (see Subsection 5.3) and that an unseen object o_k has exactly the same similarity to the prototype of the two clusters S_1 and S_2. Independently of the coefficients found by JADE, o_k will be classified as belonging to $C1$ according the cluster S_1 and as belonging to $C2$ according to S_2. Therefore, the classification for o_k is not unique.

Despite of the comment above, we think that may be this case is not so common. For this reason, we carried out experiments to check which of the two hypotheses above (**H1** or **H2**) is the most feasible. Therefore, our goal is to test how the composition of the clusters influence in the classification accuracy of JADE. The experiments described in the previous section have been carried out under hypothesis **H1**. Now we will repeat the same experiments but using clusters randomly formed. Another difference is that now we use the number of clusters we want to partition the data set as input of JADE instead of the *similarity threshold*. Moreover, now versions $V1$, $V2$ and $V3$ have no sense because all of them use a similarity threshold with the prototype of the clusters,and under **H2** the clusters are randomly formed therefore, there is no prototype. We used the following procedure on the *Iris* data set:

- Randomly select a subset of 50 objects to form the Test set.
- Training = Data set − Test set.
- Let N be the number of clusters we want to partition the Training set.
- Use JADE to compute the coefficients characterising the clusters $C_1, \ldots, C_N$.

Once all the clusters have been characterised by the coefficients given by JADE, the procedure to evaluate them is the following:

- For each *obj* in Test set.
- For each cluster C_i, use its associated coefficients for classifying *obj*. Let s_i be the proposed class.
- Let $\mathcal{S} = \{s_1, \ldots, s_N\}$ the set of classes proposed by each one of the clusters.
- If $s_i = s_j$ for all i, j then all the clusters have proposed the same classification for *obj*,
- otherwise, the classification for *obj* is the majority class.

We performed experiments with different number of clusters: N = 3, 5, and 7, and for each N we carried out 10 trials. Table 3 shows the mean of the accuracy from these 10 trials for each N. In these results we counted multiple answers as incorrect. We see that accuracy increases as N increases to achieve a stable value around the 96.0% that is similar to the accuracy obtained with versions $V1$ and $V3$ when clusters are well-formed. The *iris* data set has three solution classes. When the data set is partitioned on well-formed clusters, objects are grouped by similarity in three or four clusters. However, to obtain a similar accuracy, when the clusters are randomly formed, it is necessary to partition the data set in at least 7 clusters. Therefore a first conclusion could be that JADE performs well although clusters are randomly formed; nevertheless in this case, to obtain a comparable accuracy, it is necessary to partition the data set in more clusters than when the clusters are well formed.

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001

Table 3
Accuracies of JADE with random clusters (left part) and well-formed clusters (right part).

$N=3$	$N=5$	$N=7$	$N=9$	$V1$	$V2$	$V3$
94.2	95.6	96.4	96.0	96.6	87.8	96.2

6. Conclusions and future work

We have defined and characterised T-generable indistinguishability operators for continuous Archimedean t-norms. These operators E that can be obtained by indistinguishability operators generated by a fuzzy subset. The most interesting situation appears when the t-norm is non-strict. In this case E is T-generable when it can be obtained from crisp equivalence relations. The results of this first part have then been applied in the second part of the paper to design and implement JADE, a new method for solving classification tasks based on similarity (or indistinguishability) relations between domain objects.

JADE is based on minimising the difference between two indistinguishability relations, it is mathematically sound and the experimentation proved that it is a promising method for classification tasks. Differently than other feature selection methods, it can be used on domains with more than two solution classes. However it has an important complexity problem when the number of both attributes and objects in the training set is high. For this reason, we propose to cluster the data set and then use *JADE* on each cluster. We have experimentally seen that this is feasible in domains with either categorical or continuous attributes, and that results are comparable to the ones obtained with classifiers such as J48, Naive Bayes and Multilayer Perceptron among others. Also, we have experimented with two situations: 1) when the clusters are well-formed; and, 2) when the clusters are randomly formed. We have seen that to obtain a similar accuracy in both options, there is necessary to have a higher number of random clusters.

Acknowledgements

Authors thank to Lluís Godo his helpful comments. This research is partially funded by the projects RASO (TIN2015-71799-C2-1-P) and RPREF (CSIC Intramural 201650E044) and the grants 2014-SGR-118 and 2014-SGR-788 from the Generalitat de Catalunya.

References

[1] L. Valverde, On the structure of f-indistinguishability operators, Fuzzy Sets Syst. 17 (1985) 313–328.
[2] J. Jacas, J. Recasens, Aggregation of t-transitive relations, Int. J. Intell. Syst. 18 (2003) 1193–1214.
[3] A. Pradera, E. Trillas, E. Castiñeira, On the aggregation of some classes of fuzzy relations, in: Technologies for Constructing Intelligent Systems 2: Tools, Studies in Fuzziness and Soft Computing, Springer Verlag, New York, NY, USA, 2002, pp. 125–136.
[4] A. Pradera, E. Trillas, A note on pseudometrics aggregation, Int. J. Gen. Syst. 31 (2002) 41–51.
[5] J. Bezdek, J. Harris, Fuzzy partitions and relations; an axiomatic basis for clustering, Fuzzy Sets Syst. 1 (1978) 111–127.
[6] J. Jacas, J. Recasens, Aggregation operators based on indistinguishability operators, Int. J. Intell. Syst. 37 (2006) 857–873.
[7] E. Klement, R. Mesiar, E. Pap, Triangular Norms, Kluwer Academic Publisher, Dordrecht, The Netherlands, 2000.
[8] C.L. Walker, E. Walker, Powers of t-norms, Fuzzy Sets Syst. 129 (2002) 1–28.
[9] L. Zadeh, Similarity relations and fuzzy orderings, Inf. Sci. 3 (2) (1971) 177–200.
[10] E. Trillas, L. Valverde, An inquiry into indistinguishability operators, in: Aspects of Vagueness, Springer, Netherlands, Dordrecht, 1984, pp. 231–256.
[11] A.R. De Soto, J. Recasens, Some sets of indistinguishability operators as multiresolution families, Inf. Sci. 319 (2015) 38–55.
[12] J.R.S. Massenet, J.J. Torrens, Fuzzy implication functions based on powers of continuous t-norms, Int. J. Approx. Reason. 83 (2017) 265–279.
[13] D. Boixader, J. Recasens, Powers with respect to t-norms and t-conorms and aggregation functions, in: Fuzzy Logic and Information Fusion, Studies in Fuzziness and Soft Computing, Springer-Verlag, 2016.
[14] B.V. Dasarathy, Data mining tasks and methods: classification: nearest-neighbor approaches, in: Handbook of Data Mining and Knowledge Discovery, Oxford University Press, Inc., New York, NY, USA, 2002, pp. 288–298.
[15] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Commun. 7 (1) (1994) 39–59.
[16] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[17] K. Kira, L. Rendell, The feature selection problem: traditional methods and a new algorithm, in: AAAI, AAAI Press and MIT Press, 1992, pp. 129–134.
[18] I. Rodriguez-Lujan, R. Huerta, C. Elkan, C. Cruz, Quadratic programming feature selection, J. Mach. Learn. Res. 11 (2010) 1491–1516.

[19] L. Zhang, F. Coenen, P. Leng, An attribute weight setting method for k-NN based binary classification using quadratic programming, in: F. van Harmelen (Ed.), ECAI, IOS Press, 2002, pp. 325–329.
[20] M. Bazaraa, H. Sherali, C. Shetty, Nonlinear Programming: Theory and Algorithms, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, 1993.
[21] A. Asuncion, D. Newman, UCI machine learning repository, http://www.ics.uci.edu/~mlearn/{MLR}epository.html, 2007.
[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten.

Please cite this article in press as: E. Armengol et al., T-generable indistinguishability operators and their use for feature selection and classification, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.03.001